SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

SPYWOLF

Security Audit Report

Completed on

August 18, 2023

OVERVIEW

This audit has been prepared for **HYPE** to review the main aspects of the project to help investors make make an informative decision during their research process.

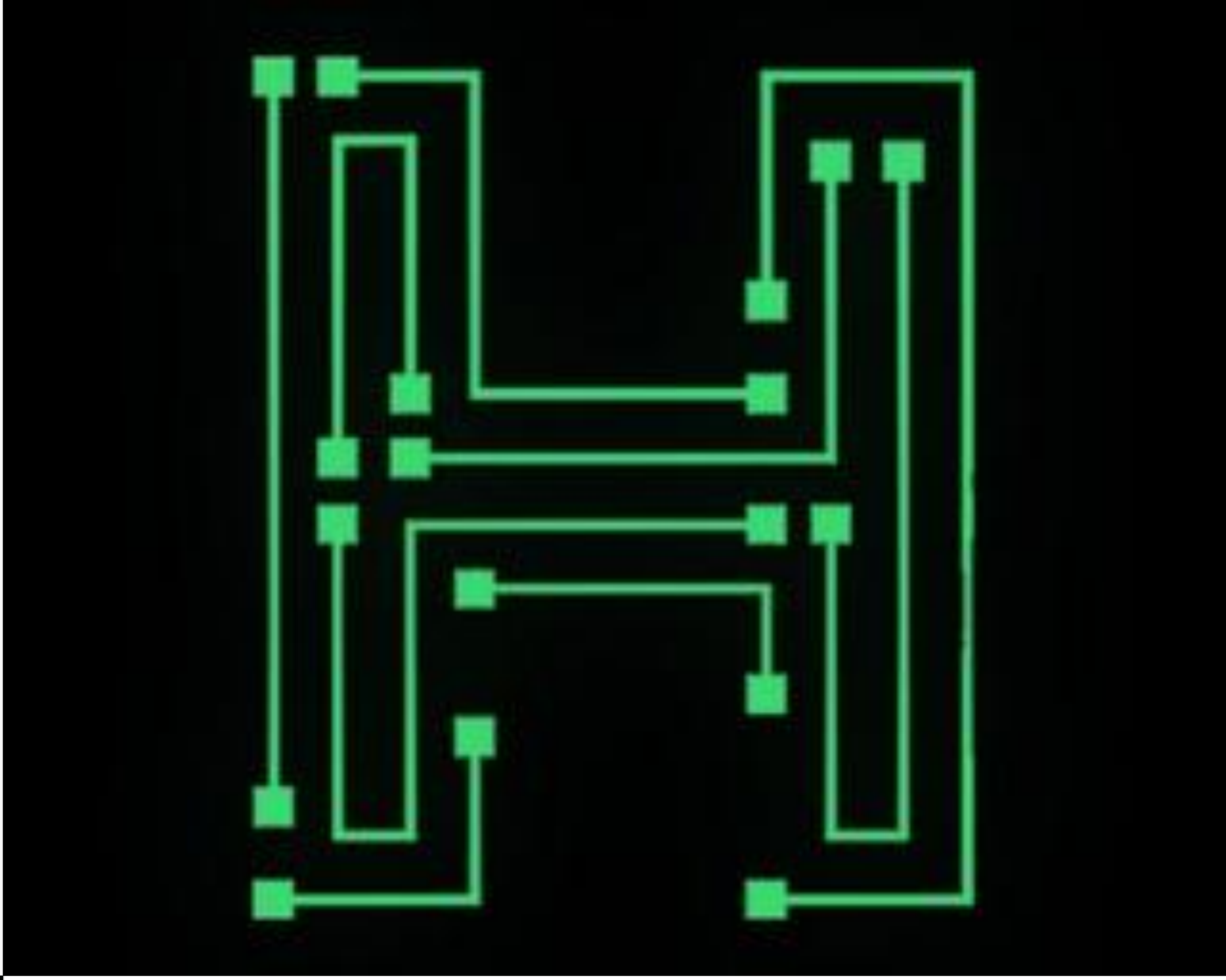
You will find a a summarized review of the following key points:

- ✓ Contract's source code
- ✓ Owners' wallets
- ✓ Tokenomics
- ✓ Team transparency and goals
- ✓ Website's age, code, security and UX
- ✓ Whitepaper and roadmap
- ✓ Social media & online presence

> *The results of this audit are purely based on the team's evaluation and does not guarantee nor reflect the projects outcome and goal*
>
> - SPYWOLF Team -

TABLE OF CONTENTS

HYPE



PROJECT DESCRIPTION

According to their website:

$HYPE is a P2P exchange capable of handling transactions for digital entities and/or fiat in exchange of crypto.

This is the first time ever this has been made as a project while also keeping customers fully anonymous.It would be similar to "localbitcoin" but without a KYC.

Release Date: Launched at August 15th, 2023

Category: Crypto Mixer

CONTRACT INFO

Token Name
TokenFarm

Symbol
N/A

Contract Address
0x0f08a5C4B670BE5bf89B8cb4ECa625d4D10b8e8A

Network
Ethereum

Language
Solidity

Deployment Date
Aug 15, 2023

Contract Type
Staking

Total Supply
N/A

Status
Launched

TAXES

Buy Tax
none

Sell Tax
none

Our Contract Review Process

The contract review process pays special attention to the following:

- ✓ Testing the smart contracts against both common and uncommon vulnerabilities
- ✓ Assessing the codebase to ensure compliance with current best practices and industry standards.
- ✓ Ensuring contract logic meets the specifications and intentions of the client.
- ✓ Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- ✓ Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

TOKEN TRANSFERS STATS

Transfer Count	N/A
Uniq Senders	N/A
Uniq Receivers	N/A
Total Amount	N/A
Median Transfer Amount	N/A
Average Transfer Amount	N/A
First transfer date	N/A
Last transfer date	N/A
Days token transferred	N/A

SMART CONTRACT STATS

Calls Count	2
External calls	2
Internal calls	0
Transactions count	2
Uniq Callers	2
Days contract called	2
Last transaction time	2023-08-18 10:59:11 UTC
Created	2023-08-15 06:59:11 UTC
Create TX	0xb4fd5d600103407480ea4abb98f3d3c455aeb63cc9ce289ae1f9c372a24ae961
Creator	0x4ab2e09b38c9798b25298b881f24e7351a84e51d

VULNERABILITY CHECK

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

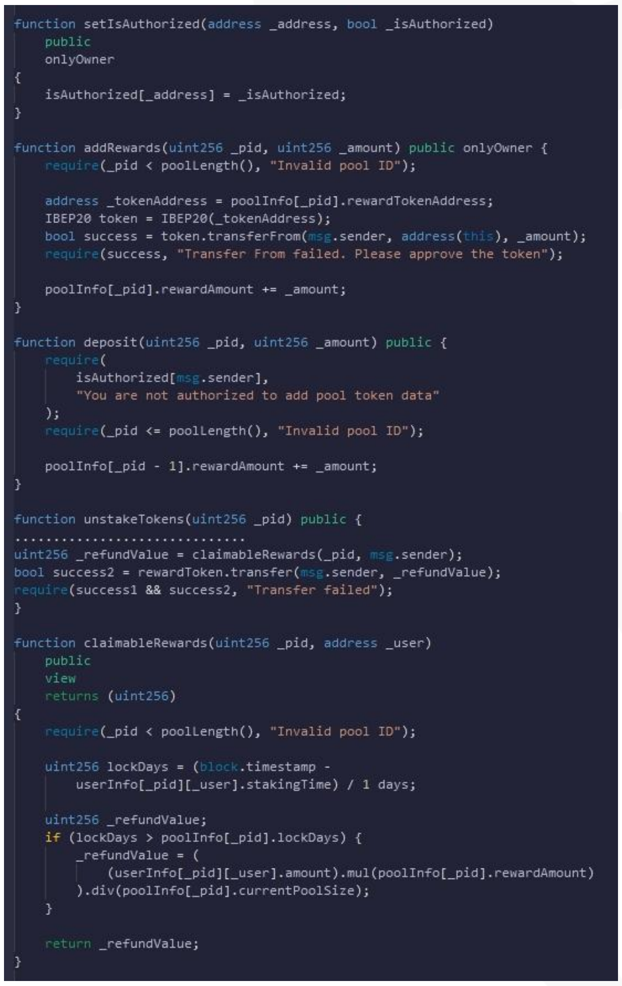
Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

⚠ High Risk

Owner can authorize address.
Owner can add rewards to each pool.
Authorized address can alter the rewardAmount's value which will only increase.
This can lead to undesired behaviour with rewards payment and cause user's investment to stuck (not enough amount to payout).



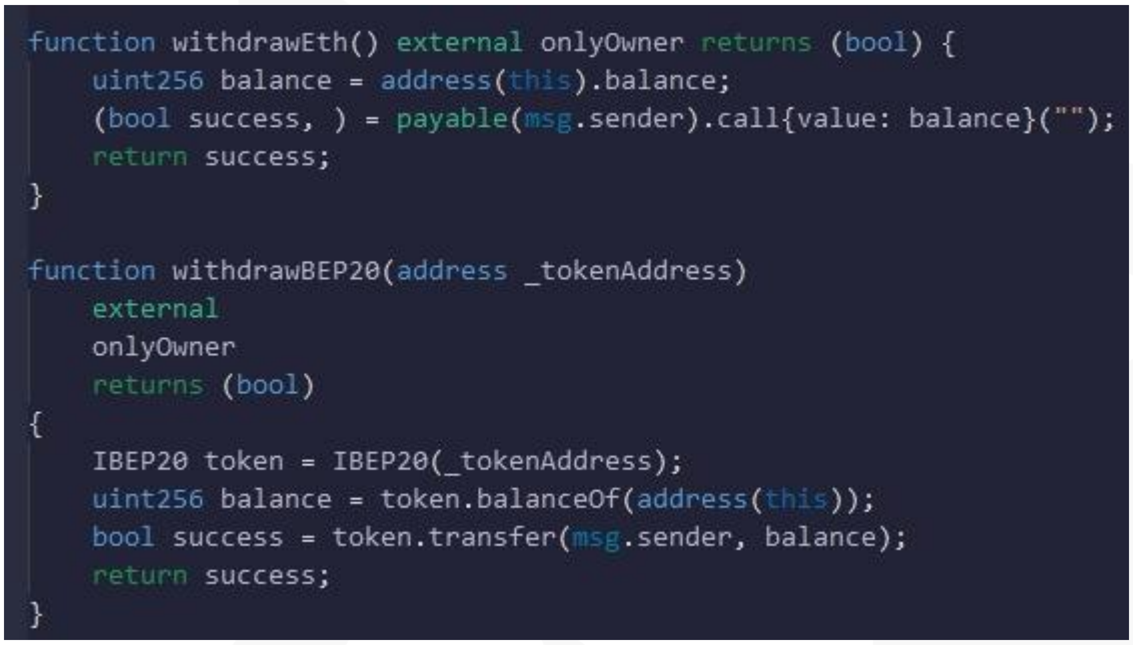
```
function setIsAuthorized(address _address, bool _isAuthorized)
    public
    onlyOwner
{
    isAuthorized[_address] = _isAuthorized;
}

function addRewards(uint256 _pid, uint256 _amount) public onlyOwner {
    require(_pid < poolLength(), "Invalid pool ID");

    address _tokenAddress = poolInfo[_pid].rewardTokenAddress;
    IBEP20 token = IBEP20(_tokenAddress);
    bool success = token.transferFrom(msg.sender, address(this), _amount);
    require(success, "Transfer From failed. Please approve the token");

    poolInfo[_pid].rewardAmount += _amount;
}

function deposit(uint256 _pid, uint256 _amount) public {
    require(
        isAuthorized[msg.sender],
        "You are not authorized to add pool token data"
    );
    require(_pid <= poolLength(), "Invalid pool ID");

    poolInfo[_pid - 1].rewardAmount += _amount;
}

function unstakeTokens(uint256 _pid) public {
..............................
uint256 _refundValue = claimableRewards(_pid, msg.sender);
bool success2 = rewardToken.transfer(msg.sender, _refundValue);
require(success1 && success2, "Transfer failed");
}

function claimableRewards(uint256 _pid, address _user)
    public
    view
    returns (uint256)
{
    require(_pid < poolLength(), "Invalid pool ID");

    uint256 lockDays = (block.timestamp -
        userInfo[_pid][_user].stakingTime) / 1 days;

    uint256 _refundValue;
    if (lockDays > poolInfo[_pid].lockDays) {
        _refundValue = (
            (userInfo[_pid][_user].amount).mul(poolInfo[_pid].rewardAmount)
        ).div(poolInfo[_pid].currentPoolSize);
    }

    return _refundValue;
}
```

- Recommendation:
  - rewardAmount’s value should only be increased with the addRewards() method, respective with the newly added tokens

FOUND THREATS

⚠ Medium Risk

*Owner can withdraw any tokens from the contract.



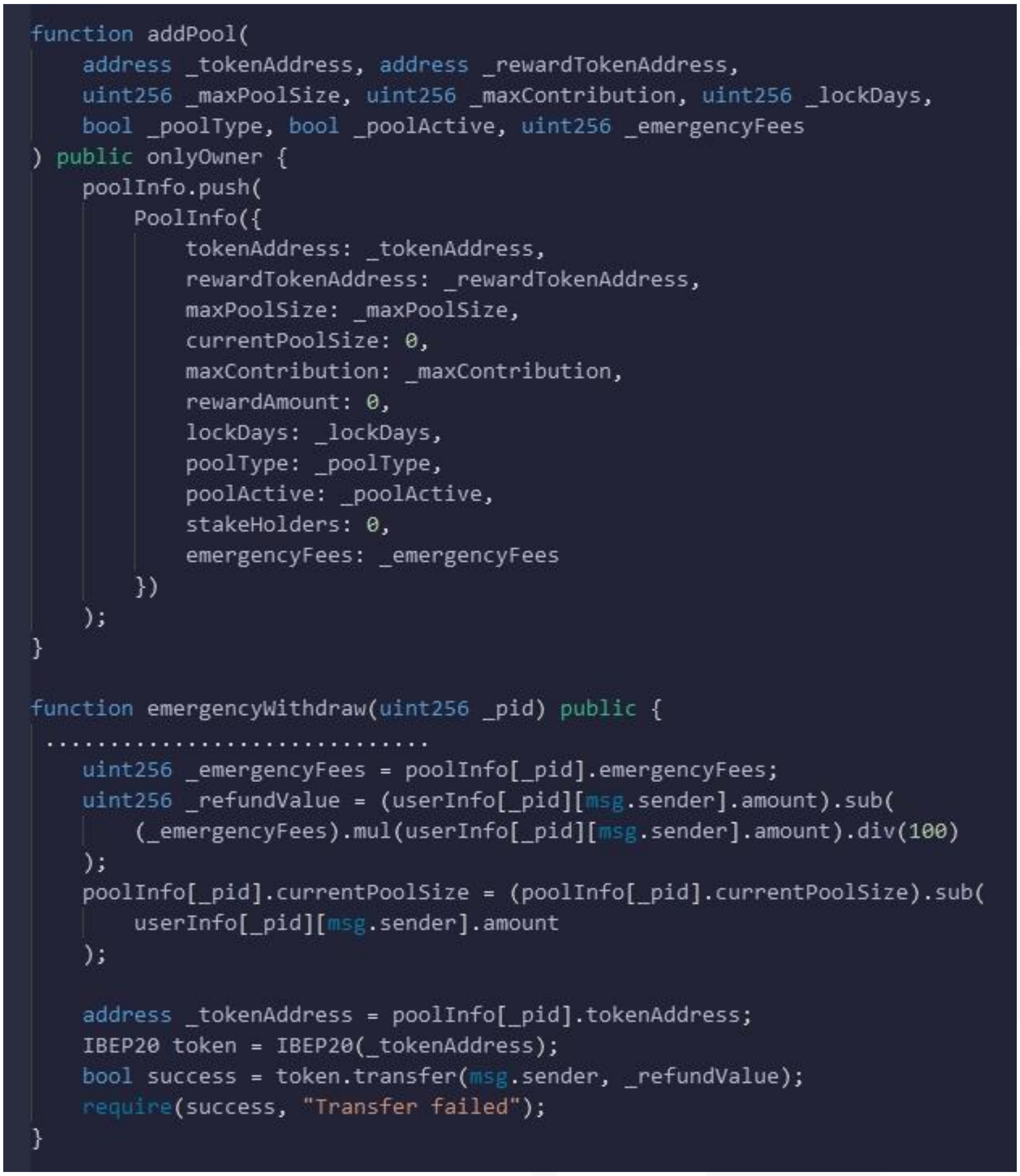
```
function withdrawEth() external onlyOwner returns (bool) {
    uint256 balance = address(this).balance;
    (bool success, ) = payable(msg.sender).call{value: balance}("");
    return success;
}

function withdrawBEP20(address _tokenAddress)
    external
    onlyOwner
    returns (bool)
{
    IBEP20 token = IBEP20(_tokenAddress);
    uint256 balance = token.balanceOf(address(this));
    bool success = token.transfer(msg.sender, balance);
    return success;
}
```

**This contract do not have receive() function and cannot receive ETH via typical transfer.*

- Recommendation:
  - As this is staking contract, tokens withdraws should be only available to investors.

Informational

Owner can create new pools.
*Owner can set emergency fees up to 100%.

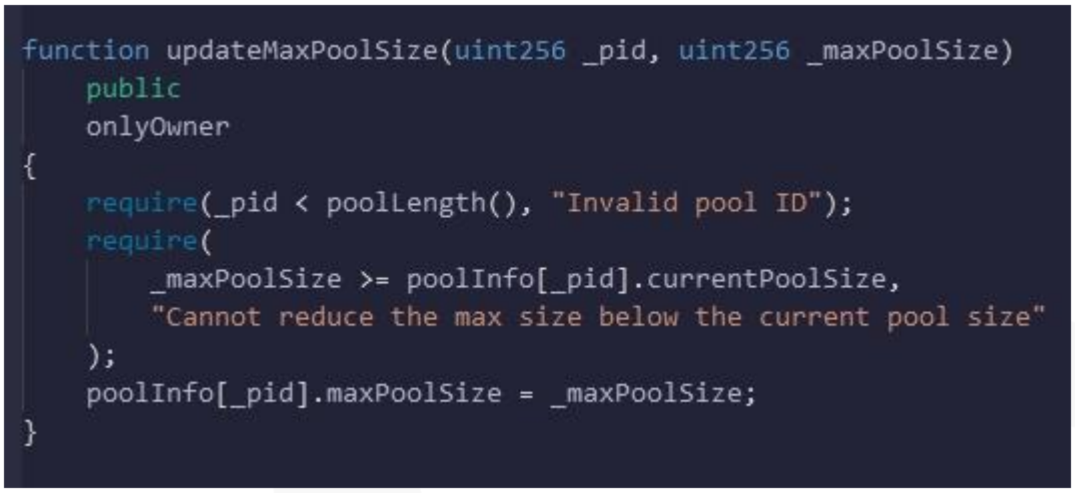


```
function addPool(
    address _tokenAddress, address _rewardTokenAddress,
    uint256 _maxPoolSize, uint256 _maxContribution, uint256 _lockDays,
    bool _poolType, bool _poolActive, uint256 _emergencyFees
) public onlyOwner {
    poolInfo.push(
        PoolInfo({
            tokenAddress: _tokenAddress,
            rewardTokenAddress: _rewardTokenAddress,
            maxPoolSize: _maxPoolSize,
            currentPoolSize: 0,
            maxContribution: _maxContribution,
            rewardAmount: 0,
            lockDays: _lockDays,
            poolType: _poolType,
            poolActive: _poolActive,
            stakeHolders: 0,
            emergencyFees: _emergencyFees
        })
    );
}

function emergencyWithdraw(uint256 _pid) public {
 ..............................
    uint256 _emergencyFees = poolInfo[_pid].emergencyFees;
    uint256 _refundValue = (userInfo[_pid][msg.sender].amount).sub(
        (_emergencyFees).mul(userInfo[_pid][msg.sender].amount).div(100)
    );
    poolInfo[_pid].currentPoolSize = (poolInfo[_pid].currentPoolSize).sub(
        userInfo[_pid][msg.sender].amount
    );

    address _tokenAddress = poolInfo[_pid].tokenAddress;
    IBEP20 token = IBEP20(_tokenAddress);
    bool success = token.transfer(msg.sender, _refundValue);
    require(success, "Transfer failed");
}
```

**For more information check the Tokenomics slide (slide 08)*

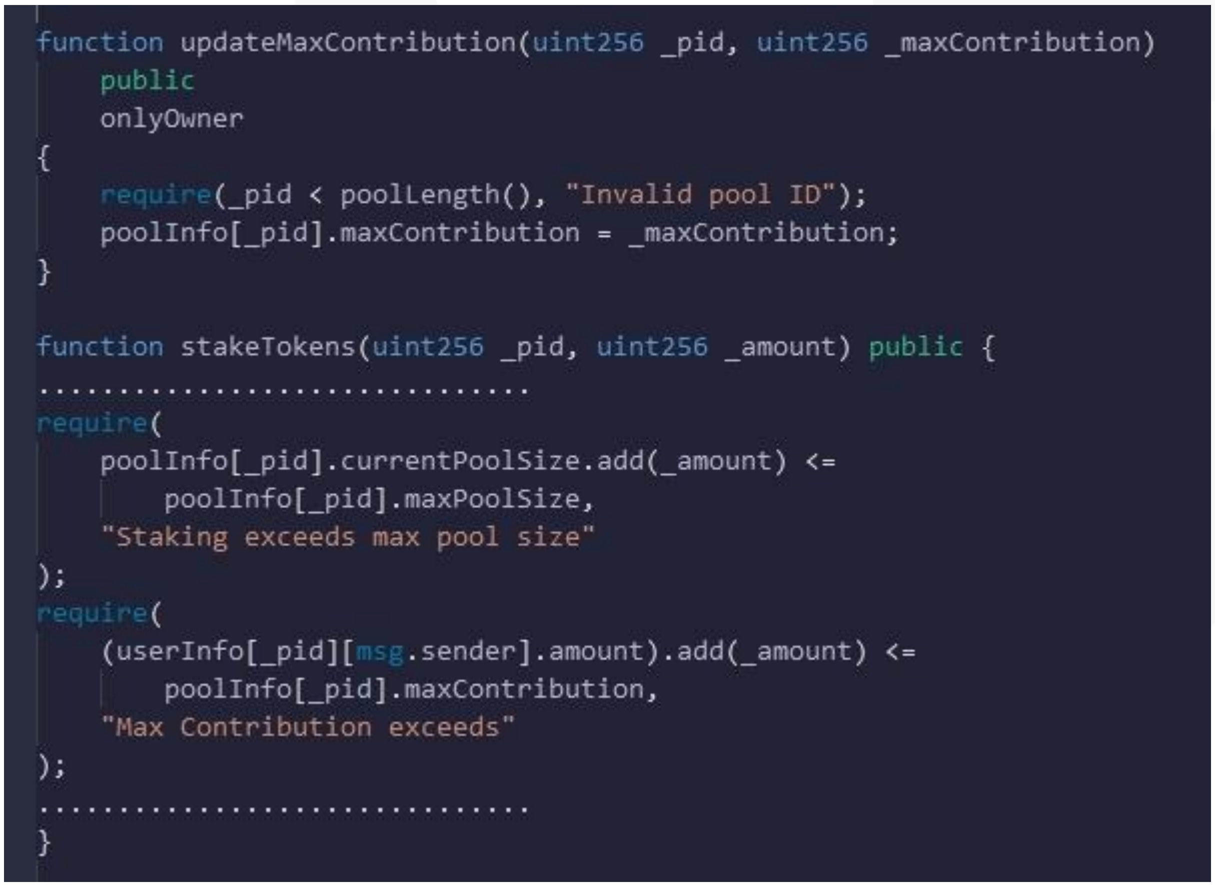
Informational

Owner can only increase the size of created pool (the total amount that can be staked in that pool).



```
function updateMaxPoolSize(uint256 _pid, uint256 _maxPoolSize)
    public
    onlyOwner
{
    require(_pid < poolLength(), "Invalid pool ID");
    require(
        _maxPoolSize >= poolInfo[_pid].currentPoolSize,
        "Cannot reduce the max size below the current pool size"
    );
    poolInfo[_pid].maxPoolSize = _maxPoolSize;
}
```

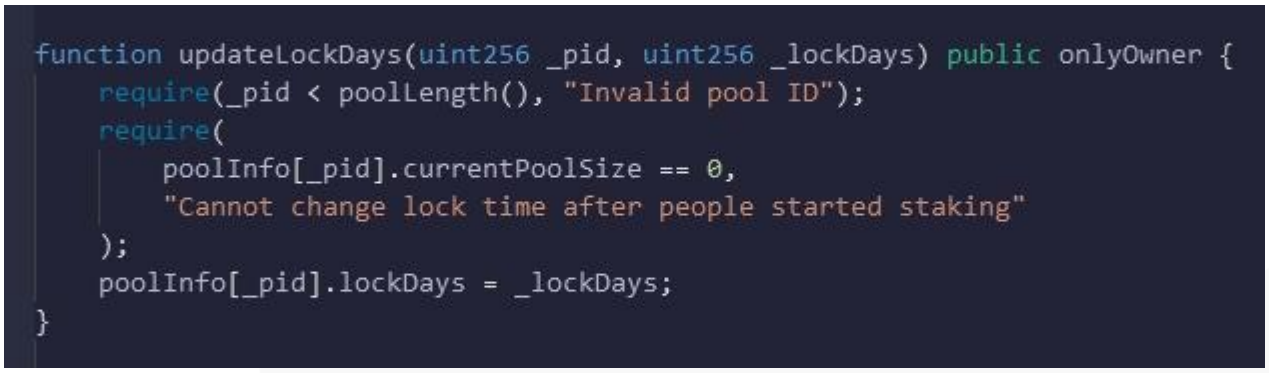
Owner can change max pool contribution (max amount that each user can stake in the pool).



```
function updateMaxContribution(uint256 _pid, uint256 _maxContribution)
    public
    onlyOwner
{
    require(_pid < poolLength(), "Invalid pool ID");
    poolInfo[_pid].maxContribution = _maxContribution;
}

function stakeTokens(uint256 _pid, uint256 _amount) public {
..............................
require(
    poolInfo[_pid].currentPoolSize.add(_amount) <=
        poolInfo[_pid].maxPoolSize,
    "Staking exceeds max pool size"
);
require(
    (userInfo[_pid][msg.sender].amount).add(_amount) <=
        poolInfo[_pid].maxContribution,
    "Max Contribution exceeds"
);
..............................
}
```

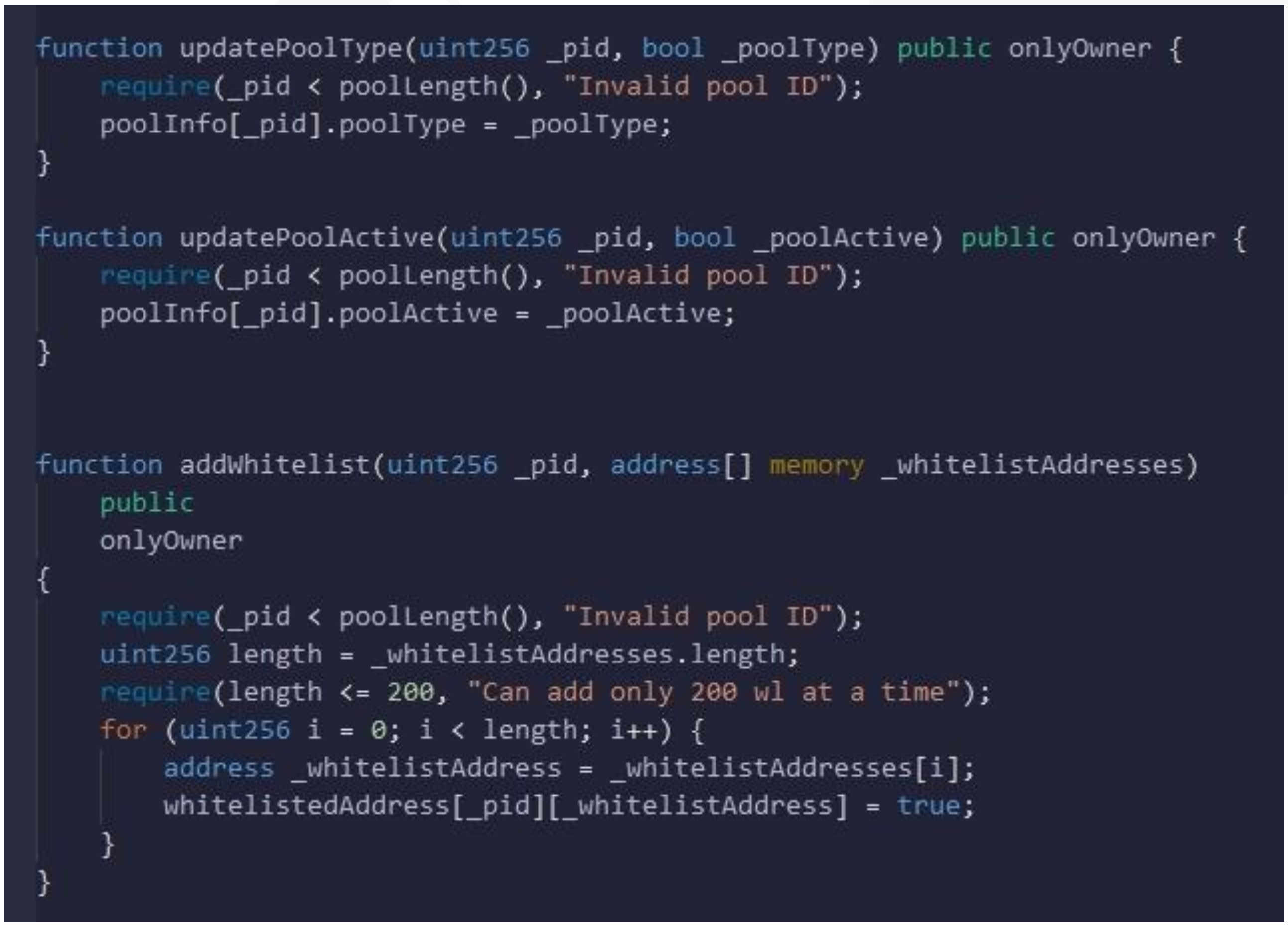
Informational

Owner can change lock period for staking pool only when the pool have 0 tokens in it (no investors for this pool yet).



```
function updateLockDays(uint256 _pid, uint256 _lockDays) public onlyOwner {
    require(_pid < poolLength(), "Invalid pool ID");
    require(
        poolInfo[_pid].currentPoolSize == 0,
        "Cannot change lock time after people started staking"
    );
    poolInfo[_pid].lockDays = _lockDays;
}
```

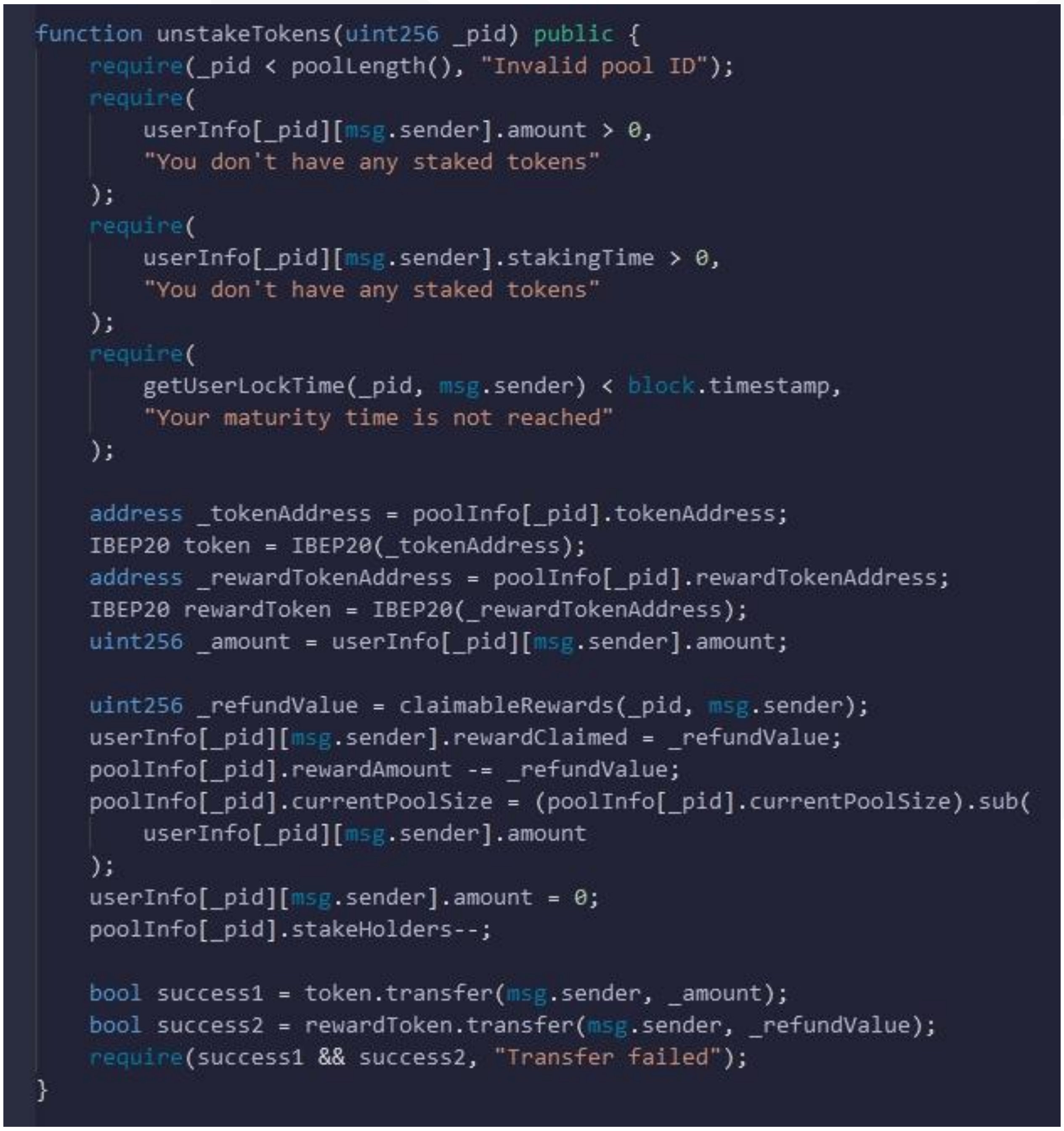
Owner can update pool type (public or private).
Owner can activate/deactivate staking pool (only applies for new depositors).
Owner can add whitelisted users that can participate in private staking pools.



```
function updatePoolType(uint256 _pid, bool _poolType) public onlyOwner {
    require(_pid < poolLength(), "Invalid pool ID");
    poolInfo[_pid].poolType = _poolType;
}

function updatePoolActive(uint256 _pid, bool _poolActive) public onlyOwner {
    require(_pid < poolLength(), "Invalid pool ID");
    poolInfo[_pid].poolActive = _poolActive;
}


function addWhitelist(uint256 _pid, address[] memory _whitelistAddresses)
    public
    onlyOwner
{
    require(_pid < poolLength(), "Invalid pool ID");
    uint256 length = _whitelistAddresses.length;
    require(length <= 200, "Can add only 200 wl at a time");
    for (uint256 i = 0; i < length; i++) {
        address _whitelistAddress = _whitelistAddresses[i];
        whitelistedAddress[_pid][_whitelistAddress] = true;
    }
}
```

Informational

Users can unstake tokens and receive rewards, only when the current pool staking period is reached.
Example - If staking pools is for 10 days, users can unstake earlier after 10 days.
If users want to pull their investment before that period, emergency fees (which can be different to every individual pool) will apply.
For more information check slide 08.



```
function unstakeTokens(uint256 _pid) public {
    require(_pid < poolLength(), "Invalid pool ID");
    require(
        userInfo[_pid][msg.sender].amount > 0,
        "You don't have any staked tokens"
    );
    require(
        userInfo[_pid][msg.sender].stakingTime > 0,
        "You don't have any staked tokens"
    );
    require(
        getUserLockTime(_pid, msg.sender) < block.timestamp,
        "Your maturity time is not reached"
    );

    address _tokenAddress = poolInfo[_pid].tokenAddress;
    IBEP20 token = IBEP20(_tokenAddress);
    address _rewardTokenAddress = poolInfo[_pid].rewardTokenAddress;
    IBEP20 rewardToken = IBEP20(_rewardTokenAddress);
    uint256 _amount = userInfo[_pid][msg.sender].amount;

    uint256 _refundValue = claimableRewards(_pid, msg.sender);
    userInfo[_pid][msg.sender].rewardClaimed = _refundValue;
    poolInfo[_pid].rewardAmount -= _refundValue;
    poolInfo[_pid].currentPoolSize = (poolInfo[_pid].currentPoolSize).sub(
        userInfo[_pid][msg.sender].amount
    );
    userInfo[_pid][msg.sender].amount = 0;
    poolInfo[_pid].stakeHolders--;

    bool success1 = token.transfer(msg.sender, _amount);
    bool success2 = rewardToken.transfer(msg.sender, _refundValue);
    require(success1 && success2, "Transfer failed");
}
```

RECOMMENDATIONS FOR

GOOD PRACTICES

1 Consider fundamental tradeoffs

2 Be attentive to blockchain properties

3 Ensure careful rollouts

4 Keep contracts simple

5 Stay up to date and track development

HYPE

GOOD PRACTICES FOUND

- ✓ The owner cannot mint new tokens after deployment
- ✓ The smart contract utilizes "SafeMath" to prevent overflows

This is staking contract.

*Emergency fees are imposed only when users want to withdraw their staked tokens before the pool's mature period.
Example - If pool's staking period is 10 days but user wants to pull their investment on day 7 -> user's investment will be subject to emergency fees.
No emergency fees are imposed on investments reached the pool's mature period.
Take into consideration that the emergency fees for each staking pool may vary and can be set up to 100%.

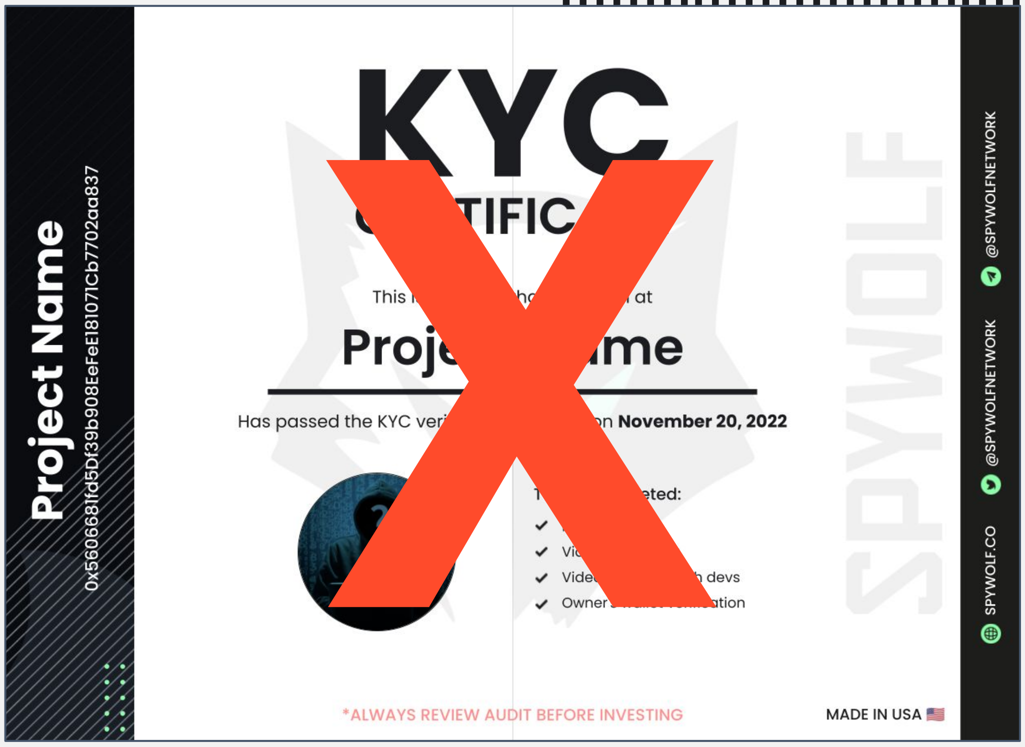
THE TEAM

⚠️ The team is annonymous

KYC INFORMATION

No KYC

We recommend the team to get a KYC in order to ensure trust and transparency within the community.



Project Name

0x5606681fd5Df39b908EeFeE181071Cb7702aa837

KYC

TIFIC

This at

Proj me

Has passed the KYC veri on **November 20, 2022**

eted:

✓

✓ Vi

✓ Vide h devs

✓ Owner ation

SPYWOLF

*ALWAYS REVIEW AUDIT BEFORE INVESTING

MADE IN USA 🇺🇸

@SPYWOLFNETWORK

@SPYWOLFNETWORK

SPYWOLF.CO

WEBSITE

Website URL
https://hype-eth.com/

Domain Registry
https://www.hostinger.com

Domain Expiration
2024-08-02

Technical SEO Test
Passed

Security Test
Passed. SSL certificate present

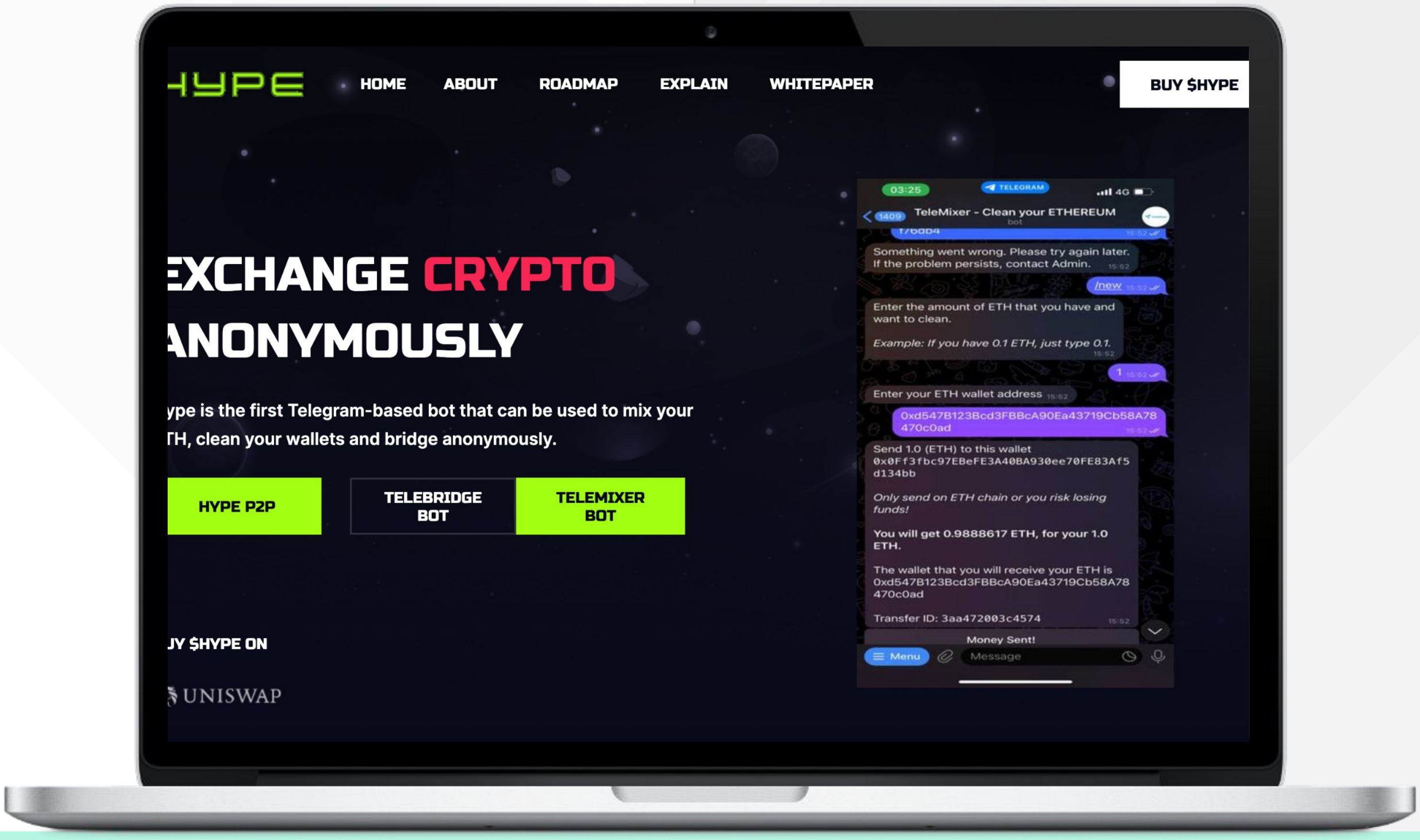
Design
Very nice design with appropriate color scheme and graphics.

Content
The information helps new investors understand what the product does right away. No grammar mistakes found..

Whitepaper
No

Roadmap
Yes, goals set without time frames.

Mobile-friendly?
Yes



EXCHANGE CRYPTO ANONYMOUSLY

ype is the first Telegram-based bot that can be used to mix your TH, clean your wallets and bridge anonymously.

TeleMixer - Clean your ETHEREUM
bot

f76db4 15:52

Something went wrong. Please try again later. If the problem persists, contact Admin. 15:52

/new 15:52

Enter the amount of ETH that you have and want to clean.

Example: If you have 0.1 ETH, just type 0.1. 15:52

1 15:52

Enter your ETH wallet address 15:52

0xd547B123Bcd3FBBcA90Ea43719Cb58A78470c0ad 15:52

Send 1.0 (ETH) to this wallet 0x0Ff3fbc97EBeFE3A40BA930ee70FE83Af5d134bb

Only send on ETH chain or you risk losing funds!

You will get 0.9888617 ETH, for your 1.0 ETH.

The wallet that you will receive your ETH is 0xd547B123Bcd3FBBcA90Ea43719Cb58A78470c0ad

Transfer ID: 3aa472003c4574 15:52

Money Sent!

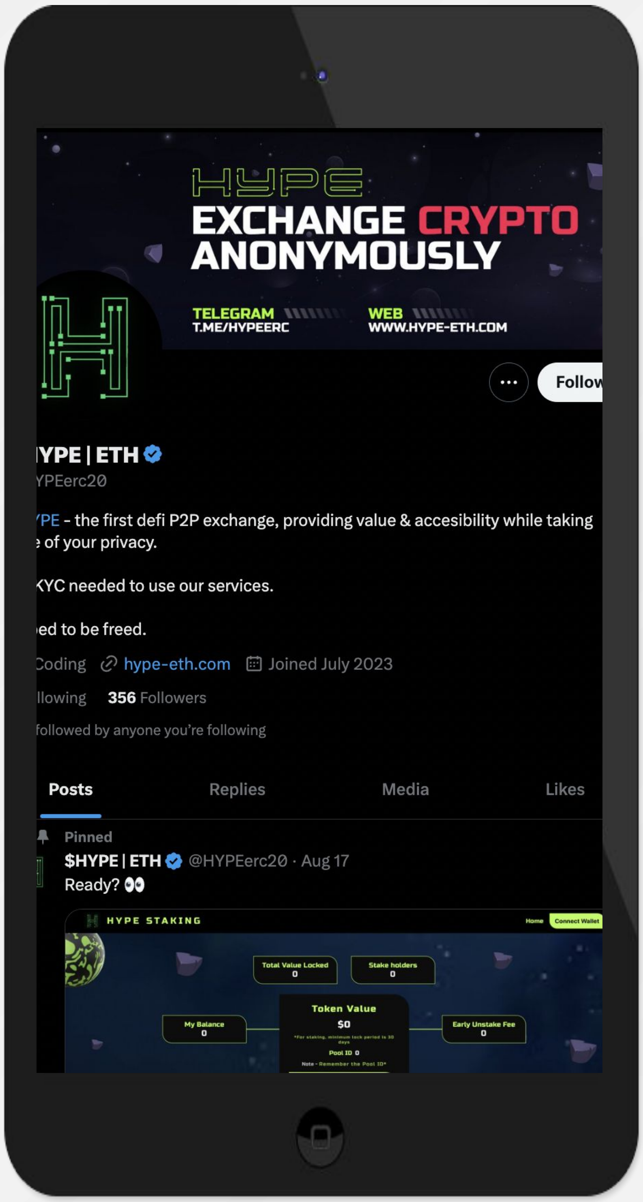
hype-eth.com

SOCIAL MEDIA

& ONLINE PRESENCE

ANALYSIS

Project's social media pages are active



HYPE
EXCHANGE CRYPTO
ANONYMOUSLY

TELEGRAM
T.ME/HYPEERC

WEB
WWW.HYPE-ETH.COM

YPE | ETH
YPEerc20

'PE - the first defi P2P exchange, providing value & accesibility while taking
: of your privacy.

KYC needed to use our services.

ed to be freed.

Coding hype-eth.com Joined July 2023

llowing 356 Followers

followed by anyone you're following

Pinned
$HYPE | ETH @HYPEerc20 · Aug 17
Ready? 👀

Twitter

@Hypeerc20

- 334 followers
- Posts frequently
- Active

Discord

- Not available

Telegram

@hypeerc

- 756 members
- Active members
- Active mods

Medium

- Not available

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- ✓ OVER 500 SUCCESSFUL CLIENTS
- ✓ MORE THAN 500 SCAMS EXPOSED
- ✓ MILLIONS SAVED IN POTENTIAL FRAUD
- ✓ PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS
- ✓ CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.

While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.

No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.

The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.

No applications were reviewed for security. No product code has been reviewed.